

Amendments to the Claims 13-52
Control Number: 10/812,998, Lee Melvin Hinman

This listing of claims will replace all prior versions, and listing, of claims in the application:

Listing of Claims:

- 13. Cancelled
- 14. Changes as noted
- 14. Old 14 a) to 14 j) now moved to 26 to 36 as New claims.
- 15. Changes as noted
- 16. Changes as noted
- 16. Old 16 a) to 16 f) now moved to 37 to 46 as New claims.
- 17. Changes as noted
- 17. Old 17 a) to 17 d) now moved to 47 to 50 as New claims.
- 18. Changes as noted
- 19. Changes as noted
- 20. Changes as noted
- 21. Changes as noted
- 22. Changes as noted
- 23. Changes as noted
- 24. Changes as noted
- 25. Changes as noted
- 26-36. New as moved from 14 a) to 14 j)
- 37-46. New as moved from 16 a) to 16 f)
- 47-50. New as moved from 17 a) to 17 d)
- 51-52. New to replace 13

Notation: Fig. 1 FLOW Chart 100, referred as (100), Fig. 2 (200)

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

51. (New) A method for forming an E-commerce Engine for quotes/orders so a buying/selling vendor connected on the internet or intranet with a web site whereas said vendor provides a series of product/service option questions are presented to the user a prospect company whereas the system retrieve the options via a process of parsing each element in an XML Schema Definition-XSD file with subelements by recursive programming for products/services that are Made to Specification of options whereas answers are required before a price can be provided where;

the prospect user selects the Product Category (100) from the screen of the buying/selling company web site;

system processes checks (101) IF an XSD file is NOT present for the Product the system goes to (115) processes it as a normal inventory item;

IF an XSD is present system processes checks IF any option questions (102), are NOT present the system goes to (115), or if product option questions are present;

the system then displays a primary screen with the list of product options for each element (103);

the user answers the option questions with multiple data types numeric/alpha numeric input/selection from a drop down field (104);

the system processes the math computations defined in XSLT (105) and also processes portal requests for quotes (113) with the user selected product option answers to receive a web services with no screens displayed as all is done via web services with the quoted/order dollar amount is returned to the portal (114);

the system processes each subelement (106) checking for errors per the defined rules described in later claims;

the system processes IF errors exists back to (102) or IF no errors proceeds on to (107) final checkout; the system writes the XMLOptions user selected data, the e-commerce order (115), and history (112);

the system also allows at user entered answers (104) IF more information exists in the subelement then (111), (also Fig. 6) the systems display a ICON and if the user clicks it another screen is displayed that allows user to enter data on the more information screen that also updates the primary screen after the user leaves each field;

the system checks the subelement IF Scalable Vector Graphics-SVG graphic exists IF NO goes to update Math (105), Or IF Yes the system uses the base information to generate the graphic image then the user enters data, then clicks the Update button (109) (also Fig. 7) which redraws the SVG image to the new input information.

52. (New) A method for defining a portal/e-Market-Places that is compatible with the e-commerce systems set forth where users can get multiple quotes for products that are Made to Specification for products/services that have options;

the user selects the Product Category (200);

then a list of buying/selling vendors is displayed for the prospect to select which ones they want to quote (201);

system processes checks (202) IF an XSD file is NOT present for the Product the system goes to (206) processes it as a normal inventory product but IF an XSD is present;

system processes checks IF any product option questions (203), is NOT present the system goes to (206), or if product option questions are present;

the system then displays a screen formatted per claim 31 and 32 with the list of product options for each element (204);

the user answers the product option questions with multiple format type numeric/alpha numeric input/selection from a drop down field (205);

the portal system processes sending the user answered product options to the selected buying/selling Vendors Co "a", Co "b", Co "c", -- Co "n" capable of receiving a call via web services that were selected in (201);

the buying/selling Vendors e-commerce system receives product option answers receive a call via web services to process each subelement (106) checking for errors per the defined rules described in later claims;

the buying/selling Vendors e-commerce system processes and IF errors exist it sends back to the Portal a NO BID or IF no errors sends back to the Portal the price;

then the users screen display the list of buying/selling vendors (207) with price;

the user selects the buying/selling vendor they want to place the order to and the system send that to the selected vendor e-commerce system then it assigns the order number and send it back to the portal where it is displayed;

the system writes the portal history and the user proceeds to the buying/selling vendors e-commerce site to complete order payment credit card information if NOT on open terms.

14. (Changes as noted) A software method for defining an E-commerce and Portal systems for product specifications utilizing product option variable questions to automatically generate a prospects user e-Commerce screen that is connected to the internet or intranet, derived from an XML Schema Definition-XSD (maintained standard by W3C) file indicated as a given implementation of just one schema and any other applicable schema's file name can be used in other implementation such as Document Type Definition-DTD files, or other new industry schemas.

a) to j)

15. (Changes as noted) The method as set forth in claim 14 wherein that by repeating the elements in the XSD to achieve unlimited elements product option specifications product option questions with recursive programming parsing each element.

16. (Changes as noted) The method as set forth in claim 14 wherein that product option selections will have user definable cross checking between XSD product option element selections; ~~the option selections having user definable cross checking between elements selections;~~ as an example

<relations>

<exact element="CouponID" value="psc#1" errormessage=" 12 Your coupon ID is not valid " />

</relations>

Fixed beginning tag <relations>,

Fixed beginning tag Said selections selected from the group consisting of Six (6) allow types consisting,

<exact element=", or <minInclusive or <minExclusive or <maxInclusive or <maxExclusive or <pattern,

the attribute for an existing element name CouponID" value=,

the attribute for the screen error message words that varies for each subelement within the product category "psc#1" errormessage=" 12 Your coupon ID is not valid ",

Fixed ending tag />,

Fixed ending tag </relations>.

a) to f)

17. (Changes as noted) A software-method as set forth in claim 51 wherein E-commerce system further processes the product option specifications for the product utilizing the XSL file the ability the Extensible Stylesheet Language Family-XSL file the ability of math to compute result fields back into the e-commerce.

a) to d)

18. (Changes as noted) The method as set forth in claim 14 wherein for File upload: of graphic image files required for the production of the product the input allows for the location of a local computer file to be uploaded to the server to a designated location;

a) **Upload URL:** Location on the web server directory via ftp, as an example

c:/webhost4life aspnet/sriflcom/www_devstore/InputProductImages/ straight directory upload, and

for ftp://PSCFTPData:LogOnID@ftp.srifl.com/DataPath/;

b) **Upload temp URL:** The temp directory is specified as to where the file is placed until final check then it is put in the next location, ~~as an example~~

c:/webhost4life/aspnet/sriflcom/www_devstore/InputProductImages/temp.

19. (Changes as noted) The method as set forth in claim 14 wherein for indication of an element being a SVG graphic type ~~as an example~~ Fixed beginning tag <isSVGFile>, the attribute true or false, Fixed ending tag </isSVGFile>, of what can be achieved, with the aspect to dynamically integrate the original master SVG graphic while linking the graphics to product option specific related items is the pricing ability, real time new graphic design display ability, and saving the results in a single database table row along with the XML options selected.

20. (Changes as noted) The method as set forth in claim 14 wherein for order final checkout will NOT occur until update cart has been successfully completed without any ~~errors-on errors process (106)~~ as checked in claims 33, 34, 37, 39, 41, 43, 45, and 46 on any of the elements for processing integrity.

21. (Changes as noted) The method as set forth in claim 14 wherein for each product option is contained in an element definitions for standalone and for the Portal access additional subset <xs:annotation> within each element of the e-commerce system, allowing in the Portal the prospects to select by the subset product options without forcing in house namespace labels ~~as an example~~ ;

beginning tag <m2specsXMLElID>,
the attribute 'QuoteOrderType',
Fixed ending tag </m2specsXMLElID>.

22. (Changes as noted) The method as set forth in claim 13 52 wherein for a system defined allowed number of suppliers that a prospect can select to provide quotes via Web Services.

23. (Changes as noted) The method as set forth in claim 13 52 wherein for the returned quoted price via web services is displayed to the prospects screen with the ability to select the supplier/buyer to award the contract/seller.

24. (Changes as noted) The method as set forth in claim 13 51 wherein the portal mode that allows for the prospect continues on to the awarded supplier's e-commerce site for completion of credit processing information to consummate the commerce transaction.

25. (Changes as noted) The method as set forth in claim 13 51 wherein for the portal will retain element product options selected history information of the transactions.

26. (New) The method as set forth in claim 14 wherein the process parses each element in the XSD and this subelement declarations of Title for data input to be displayed on the screen for each of the XML Schema Definition Language-XSD with the;

Fixed beginning tag <title>,
the attribute for the screen title words that varies for each element within the product category '1 Width Inches',
Fixed ending tag </title>.

27. (New) The method as set forth in claim 14 wherein the process parses each element in the XSD and the More information Icon with the URL links to allow opening of another screen that the site administrator can have complete control of the layout while still transferring product option selection information to the main screen for each of the XML Schema Definition Language-XSD subelements with the;

Fixed beginning tag <helpURL>,
the attribute URL that varies for each subelement within the product category
'MoreInfo/M2specsXML339950-2201-001.html#width',
Fixed ending tag </helpURL>.

28. (New) The method as set forth in claim 14, 27 wherein the process parses each element in the XSD and The help width of the screen for for each of the XML Schema Definition Language-XSD subelements with the;

Fixed beginning tag <helpwidth>,
the attribute help width that varies for each subelement within the product category '700',
Fixed ending tag </helpwidth>.

29. (New) The method as set forth in claim 14, 27 wherein the process parses each element in the XSD and The help height of the screen for each of the XML Schema Definition Language-XSD subelements with the;

Fixed beginning tag <helpheight>,
the attribute help height that varies for each subelement within the product category '350',
Fixed ending tag </helpheight>.

30. (New) The method as set forth in claim 14, 27 wherein the process parses each element in the XSD and _The Help title of the new screen for each of the XML Schema Definition Language-XSD subelements with the;

Fixed beginning tag <helptitle>,
the attribute Help title that varies for each subelement within the product category 'More Info For 1 Width Window Inches',
Fixed ending tag </helptitle>.

31. (New) The method as set forth in claim 14 wherein the process parses each element in the XSD and this Format ability of Displaying on the prospects screen the prompt questions, input, and more information link on the following three conditions as to how many are displayed on a line of the screen with the;

Fixed beginning tag <prompt>,
the attribute format that varies for each subelement within the product category 'SameLine', all on the same line is default,
other recognized attributes,
the prompt is on one line and input/more information link on the next line 'NewLine', or
the prompt and input/more information link is on the same line 'Normal',
Fixed ending tag </prompt>.

32. (New) The method as set forth in claim 31 wherein the process parses each element in the XSD and the ability to format from a system parameter to specify percentages of the line each screen subelement component will occupy for 14, and 26 to 31;

The "40:60/40:50:10" notation means "if there is no more info, make a 40% prompt question and 60% option answer entry;
if there *is* a more info link, make a 40% prompt, then 50% option question, then 10% more info', similarly for the same line case, where there are four/five numbers instead of two/three, because there are two prompt/options on a line, and for the NewLine case, when there is no width for the prompt string, just for the option (and if present the more info link);

Fixed beginning tag <add key=,
Three Keywords ability "PSC_ColSpan_Normal" or "PSC_ColSpan_SameLine" or
"PSC_ColSpan_NewLine",
With respective user applicable value="50:50/40:50:10" or "25:25:25:25/20:25:20:25:10" or
"100/90:10",
Fixed ending tag />.

33. (New) The method as set forth in claim 14 wherein the process parses each element in the XSD and the type of data is identified for each product option element input to determine if errors in data entry occur;

Fixed beginning tag <xss:restriction,
base="xs:integer" or "string" or "decimal", and any allowed type in XSD,
Fixed ending tag >.

34. (New) The method as set forth in claim 14 wherein the process parses each element in the XSD and this Edit rules checking types on the allowed user answers;

Fixed beginning tag <xss:,
The 5 allowed types of; Low or minInclusive or maxInclusive or maxLength or pattern or any allowed XSD type,
the attribute (administrator defined for element) value="6" (any numeric value) or "[0-9]+(\.[0-9]?)?" for pattern,
Fixed ending tag />.

35. (New) The method as set forth in claim 14 wherein the process parses each element in the XSD and this Error message words if input errors occur in claim 34;

Fixed beginning tag <errormessage>,
The administrators defined attribute of the displayed words if the error occurs '1 Width Must be Minimum 6',
Fixed ending tag </errormessage>.

36. (New) The method as set forth in claim 14 wherein the process parses each element in the XSD and this Parameter to indicate if the price of an product options drop down is displayed to the prospect;

Fixed beginning tag <display-price>,
The administrators defined attribute 'True' or 'False',
Fixed ending tag </display-price>.

37. (New) The method as set forth in claim 16 wherein the process parses each element in the XSD and this type **minInclusive - a low boundary limit which may be reached**.

38. (New) The method as set forth in claim 37 wherein the process parses each element in the XSD and this type minInclusive element is used to specify a minimum valid value for another element of the XML document, this applies to both numeric values and enumerations; in the latter case, the acceptable range is the specified value and all following it in the definition of the enumeration.

39. (New) The method as set forth in claim 16 wherein the process parses each element in the XSD and this type **minExclusive - a low boundary limit which may not be reached**.

40. (New) The method as set forth in claim 39 wherein the process parses each element in the XSD and this type minExclusive element is used to specify a minimum value for another element of the XML

document, but the value specified is NOT acceptable, this applies to both numeric values and enumerations; in the latter case, the acceptable range consists of all values following the specified one in the definition of the enumeration.

41. (New) The method as set forth in claim 16 wherein the process parses each element in the XSD and this type **maxInclusive - a high boundary limit which may be reached**.

42. (New) The method as set forth in claim 41 wherein the maxInclusive element is used to specify a maximum valid value for another element of the XML document, this applies to both numeric values and enumerations; in the latter case, the acceptable range is the specified value and all preceding it in the definition of the enumeration.

43. (New) The method as set forth in claim 16 wherein the process parses each element in the XSD and this **maxExclusive - a high boundary limit which may not be reached**.

44. (New) The method as set forth in claim 43 wherein the process parses each element in the XSD and The maxExclusive element is used to specify a maximum value for another element of the XML document, but the value specified is NOT acceptable, this applies to both numeric values and enumerations; in the latter case, the acceptable range consists of all values preceding the specified one in the definition of the enumeration.

45. (New) The method as set forth in claim 16 wherein the process parses each element in the XSD and this **exact - an exact value**,

the exact element is used to specify an exact value for another element of the XML document, this applies to all non-composite XSD types: numeric types, strings, and enumerations.

46. (New) The method as set forth in claim 16 wherein the process parses each element in the XSD and this **pattern - a regular expression pattern match**,

the pattern element is used to specify a regular expression pattern that another element of the XML document should match, this applies to all XSD types derived from the string type.

47. (New) The method as set forth in claim 17 wherein the process parses each line in the XSLT for the final Selling price the company is calculating for the prospect;

Fixed beginning tag <price>,

Fixed 'xsl:value-of select/>',

The administrators defined attribute of any variable already defined with further math ability
="\$FinalPriceMin ", and

Fixed ending tag </price>.

48. (New) The method as set forth in claim 17 wherein the process parses each line in the XSLT for the Shipping cost;

Fixed beginning tag <freight>,

The administrators defined attribute notes 'type="Shipping is Selected mode in \$persqft ">,

Fixed 'xsl:value-of select/>',

The administrators defined attribute of any variable already defined with further math ability
="\$FinalShipMin ", and
Fixed ending tag </freight>.

49. (New) The method as set forth in claim 17 wherein the process parses each line in the XSLT for the Sales tax;

Fixed beginning tag <tax,
The administrators defined attribute notes 'type="Only TAX if FL ">',
Fixed <xsl:value-of select=,
The administrators defined attribute of any variable already defined with further math ability
"(\$SalesTax_price div 100) * \$FinalPriceMin * \$productqty" />,
Fixed ending tag </tax>.

50. (New) The method as set forth in claim 17 wherein the process parses each line in the XSLT for Other fields that can have math computations applied;

Fixed beginning tag <xsl:variable name=,
The administrators defined attribute of the result variable name new or already defined name
"productDiscount",
the math ability 'select=" \$basePrice - (\$basePrice * \$productqtyDiscount) "',
Fixed ending tag '>'.